

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Konektory pro sociální sítě**  
**Social Networks Connectors**

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

## Zadání bakalářské práce

Student: **Aleš Roszoha**  
Studijní program: B2647 Informační a komunikační technologie  
Studijní obor: 2612R025 Informatika a výpočetní technika  
Téma: **Konektory pro sociální sítě**  
**Social Networks Connectors**

### Zásady pro vypracování:

Cílem práce je vytvořit propojení komunitního systému, který je vyvíjen společností SCOVECO, s.r.o., se sociálními sítěmi (Twitter, G+, LinkedIn). Systém je v současné době propojen se sítí Facebook.

1. Seznamte se s požadavky systému.
  2. Seznamte se s architekturou systému.
  3. Naprogramujte a otestujte konektory do dalších sociálních sítí, dle existujícího konektoru pro Facebook.
  4. Naprogramujte podporu pro další sociální sítě v klientské části aplikace, která běží na OS Android.
- Omezení, která je nutno respektovat:
1. Konektory vytvořte pro platformu PHP5 na serveru.
  2. Funkčnost zdrojových kódů zveřejněných v rámci závěrečné práce bude demonstrována na vytvořených testech, které budou součástí výstupu. Tyto testy budou sloužit pro demonstraci funkčnosti řešení.
  3. Řešení bude plně a bez úprav integrovatelné se stávajícím systémem.

### Seznam doporučené odborné literatury:


- [1] MEDNIEKS, Z., DORNIN, L., MEIKE, G.B. AND NAKAMURA, M. Programming Android: Java Programming for the New Generation of Mobile Devices. Second Edition ed.: O'Reilly Media, 2012. 566 p. ISBN 1449316646.
- [2] ZANDSTRA, M. PHP 5 Objects, Patterns, and Practice. 1 ed.: Apress, 2005. 438 p. ISBN 1590593804.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. Jan Kožusznik, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013

  
doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



  
prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

## Prehlásenie študenta

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostrave dňa: 1.5.2013

...*Rozsaha*...  
podpis študenta

## Prehlásenie spolupracujúceho zástupcu

„Súhlasím so zverejnením tejto bakalárskej práce podľa požiadaviek čl. 26, odst. 9 Študijného a skúšobného poriadku pre štúdium v bakalárskych programoch VŠB-TU Ostrava.“

V Ostrave dňa: 1.5.2013

...*Košovec Jan*...  
podpis zástupcu

## Pod'akovanie

Rád by som poďakoval Ing. Jánovi Kožusznikovi, Ph.D. za odbornú pomoc a konzultáciu pri vytváraní bakalárskej práce. Taktiež by som chcel poďakovať mojej rodine a blízkym priateľom. Vytvorili mi výborné podmienky pre moje štúdium a realizáciu bakalárskej práce. Napokon v neposlednom rade ďakujem všetkým ľuďom z internetových diskusií, ktorí mi pomohli nájsť riešenie na mnohé problémy pri implementácii.

## **Abstrakt**

Táto práca sa zameriava na prepojenie komunitného systému, ktorý je vyvíjaný spoločnosťou SCOVECO, s.r.o., so sociálnymi sieťami Twitter, Google Plus a LinkedIn. Práca taktiež popisuje použité programovacie jazyky, autorizačné protokoly a najpopulárnejšie sociálne siete súčasnosti.

## **Kľúčové slová**

PHP, Android, OAuth, sociálne siete, Facebook, Twitter, Google Plus, LinkedIn, bakalárska práca

## **Abstract**

The main focus of this thesis is to interconnect social networks Twitter, Google Plus and LinkedIn with already created community system developed by SCOVECO, s.r.o., company. Thesis also describes used programming languages, authorization protocols and the most favorite social networks of these days.

## **Key words**

PHP, Android, OAuth, social networks, Facebook, Twitter, Google Plus, LinkedIn, bachelor thesis

## Zoznam použitých skratiek

Skratka	Význam
<b>API</b>	Application Programming Interface
<b>DNS</b>	Domain Name System
<b>DVM</b>	Dalvik Virtual Machine
<b>JSON</b>	JavaScript Object Notation
<b>JVM</b>	Java Virtual Machine
<b>GUI</b>	Graphical User Interface
<b>HTTP</b>	Hypertext Transfer Protocol
<b>PDA</b>	Personal Digital Assistant
<b>PHP</b>	Hypertext Preprocessor
<b>SDK</b>	Software Development Kit
<b>SMS</b>	Short Message Service
<b>SMTP</b>	Simple Mail Transfer Protocol
<b>SOAP</b>	Simple Object Access Protocol
<b>REST</b>	Representational State Transfer
<b>UML</b>	Unified Modeling Language
<b>URI</b>	Uniform Resource Identifier
<b>URL</b>	Uniform Resource Locator
<b>XML - RPC</b>	XML – Remote Procedure Call

---

# Obsah

1	Úvod .....	1
2	Aplikácia MeetMe .....	2
3	Architektura .....	3
3.1	Klient – Server .....	3
3.2	Klient – Server na aplikácii MeetMe .....	4
3.3	Serverová časť .....	4
3.3.1	Opis a funkcie .....	4
3.3.2	PHP .....	4
3.3.3	Apache .....	4
3.4	Klientská časť .....	5
3.4.1	Opis a funkcie .....	5
3.4.2	Java .....	5
3.4.3	Android .....	6
3.4.4	Protokol XML – RPC .....	6
4	Sociálne siete .....	7
4.1	Čo je to sociálna sieť ? .....	7
4.2	Ich prínos a ako ovplyvnili Internet .....	7
4.3	Príklady sociálnych sietí .....	8
4.3.1	Facebook .....	8
4.3.2	Google Plus .....	9
4.3.3	Twitter .....	11
4.3.4	LinkedIn .....	12
5	Prepojenie sociálnych sietí a aplikácií .....	14
5.1	Dôvody .....	14
5.2	API .....	14
5.3	Protokol OAuth .....	14



5.4	Prepojenie s MeetMe .....	16
6	Implementácia konektorov do aplikácie MeetMe .....	17
6.1	Priebeh komunikácie medzi MeetMe a sociálnymi sieťami.....	17
6.2	Opis dôležitých častí na strane serveru .....	18
6.2.1	Rozhranie ISNSConnector .....	18
6.2.2	Dôležité objekty, s ktorými pracujú konektory .....	19
6.2.3	Súbory twitter_functions.php, gplus_functions.php a linkedin_functions.php ....	20
6.3	Opis dôležitých častí na strane klienta .....	21
6.3.1	Rozhranie ISocialServiceWrapper .....	21
6.3.2	Dôležité objekty, s ktorými pracujú wrappery .....	21
6.4	Google Plus konektor a wrapper .....	22
6.4.1	Princíp fungovania.....	22
6.4.2	Google Plus API.....	22
6.4.3	Funkcie v konektore .....	23
6.4.4	Funkcie vo wrapperi .....	23
6.4.5	Problémy pri implementácii .....	24
6.5	Twitter konektor a wrapper .....	24
6.5.1	Princíp fungovania.....	24
6.5.2	Twitter API.....	25
6.5.3	Funkcie v konektore .....	25
6.5.4	Funkcie vo wrapperi .....	26
6.6	LinkedIn konektor a wrapper .....	26
6.6.1	Princíp fungovania.....	26
6.6.2	LinkedIn API .....	27
6.6.3	Funkcie v konektore .....	27
6.6.4	Funkcie vo wrapperi .....	28
6.6.5	Problémy pri implementácii .....	28
7	Záver.....	29

8	Referencie.....	30
9	Prílohy .....	34
9.1	Popis registrácie aplikácie a stiahnutia Google API.....	34
9.2	Popis registrácie aplikácie a stiahnutia Twitter API.....	35
9.3	Popis registrácie aplikácie a stiahnutia LinkedIn API.....	35

---

# 1 Úvod

Žijeme v dobe Internetu a sociálnych sietí. Každý deň ľudia píšu o svojich zážitkoch, hľadajú si nových priateľov alebo zdieľajú rôzne fotografie z osláv. Internet im otvára úplne nové možnosti komunikácie, kedy je úplne jedno či komunikujú ľudia z rovnakého štátu alebo každý z nich je na opačnej strane zemegule. Ďalej je úplne jedno akého sú pohlavia, rasy alebo vierovyznania.

Tento trend sociálnych sietí môžeme vidieť aj vo vývoji aplikácií. V dnešnej dobe je úplne bežná požiadavka prepojenie rôznych sociálnych sietí s aplikáciou, hlavne takou ktorá beží na platforme Android alebo iOS. Tento druh prepojenia je hlavnou náplňou mojej bakalárskej práce.

Klientská aplikácia MeetMe vyvíjaná firmou SCOVECO s.r.o. je momentálne prepojená s najpopulárnejšou sociálnou sieťou súčasnosti Facebookom. Primárnou funkciou tejto aplikácie je vytváranie rôznych udalostí a následne pozývanie svojich priateľov na ne. Aplikácia funguje na platforme Android a je primárne určená pre mobilné telefóny.

Cieľom mojej bakalárskej práce je vytvoriť konektory pre ďalšie populárne sociálne siete súčasnosti, konkrétne Twitter, Google Plus a LinkedIn, s aplikáciou MeetMe. Implementácia bude rozdelená na dve väčšie časti. Prvá časť sa bude týkať naprogramovania konektorov na strane serveru. Druhá časť bude zahrňovať naprogramovanie wrapperov pre jednotlivé sociálne siete v klientskej časti aplikácie, ktorá funguje na platforme Android.

---

## 2 Aplikácia MeetMe

MeetMe je aplikácia, pomocou ktorej môže užívateľ komunikovať s viacerými sociálnymi sieťami pomocou jedného účtu. Môže napríklad vytvárať nové udalosti a následne do nich pozývať svojich priateľov zo všetkých sociálnych sietí, ktoré sú prepojené s aplikáciou.

MeetMe využíva architektúru klient – server. V úlohe klienta je aplikácia na platforme Android, ktorá komunikuje s webovým serverom Apache pomocou protokolu XML-RPC. Serverová časť je napísaná v programovacom jazyku PHP a klientská časť v programovacom jazyku Java.

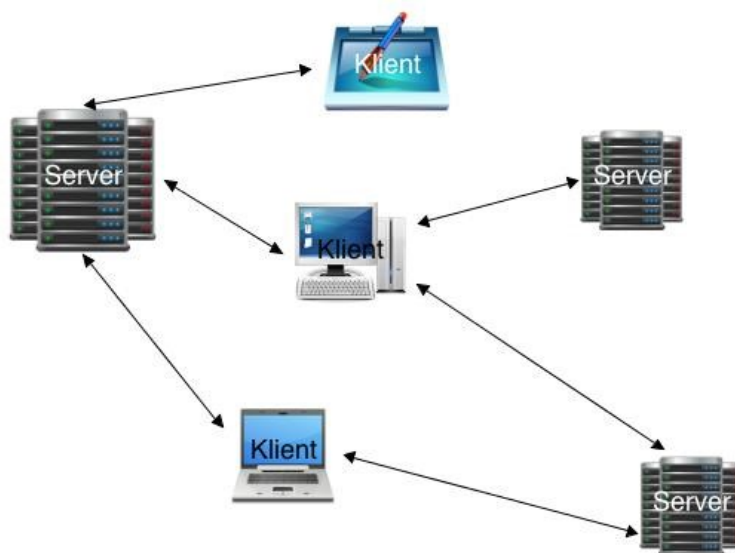
Nato aby sme mohli túto aplikáciu používať potrebujeme telefón, ktorý beží na operačnom systéme Android minimálne na verzii 2.2 a mať vytvorený účet na niektorej z najpopulárnejších sociálnych sietí. Momentálne je podporovaný iba Facebook. Po dokončení tejto bakalárskej práce pribudne podpora sociálnych sietí Twitter, Google Plus, LinkedIn.

---

## 3 Architektúra

### 3.1 Klient – Server

Klient – server je sieťová architektúra, ktorá oddeľuje od seba dve aplikácie. Klient je často aplikácia s grafickým rozhraním. Táto aplikácia žiada o určité služby druhú aplikáciu nazývanú server. Úlohou serveru je poskytovať služby klientským aplikáciám. Na tomto modeli je založený napríklad Email, Web alebo pripojenie do databáze. Ďalej sa používa vo väčšine obchodných alebo firemných aplikáciách. Taktiež protokoly ako HTTP, SMTP alebo DNS používajú tento model. Najčastejší klienti bývajú webové prehliadače a v úlohe serverov to bývajú najmä rôzne webové, databázové alebo emailové servery. [1]



Obrázok 3.1: Obrázok architektúry klient – server

#### Charakteristiky klienta:

- aktívny
- posiela žiadosti serveru
- čaká a dostáva odpovede
- väčšinou je pripojený súčasne k malému počtu serverov
- väčšinou komunikuje priamo s koncovými užívateľmi, pomocou GUI

---

### **Charakteristika serveru:**

- pasívny
- načúva na sieti a reaguje na žiadosti pripojených, autorizovaných klientov
- po prijatí požiadavku obslúži klienta
- môže vzdialene inštalovať/odinštalovať aplikácie a prenášať dáta ku klientom [1]

## **3.2 Klient – Server na aplikácii MeetMe**

### **3.3 Serverová časť**

#### **3.3.1 Opis a funkcie**

Túto časť tvorí webový server Apache, ktorý komunikuje s klientskou aplikáciou a sociálnymi sieťami.

Všetka aplikačná logika je napísaná v skriptovacom jazyku PHP. Na tento server klientská aplikácia posiela rôzne požiadavky. Server spravuje požiadavky z klientskej aplikácie, zasiela získané dáta zo sociálnych sietí klientskej aplikácii a komunikuje s databázou.

#### **3.3.2 PHP**

PHP je open source programovací jazyk, ktorý sa používa na strane serveru a je primárne určený na programovanie dynamických webových stránok. PHP však možno použiť aj na tvorbu konzolových a desktopových aplikácií. Syntax jazyka sa inšpirovala programovacími jazykmi ako C, Java alebo Perl. Hlavným cieľom tohto jazyka je rýchle vytváranie dynamických stránok. Skripty napísané v tomto jazyku sa vykonávajú na strane serveru a klient dostáva len výsledok. PHP podporuje veľa knižníc pre rôzne účely ako sú napríklad spracovanie textu, grafiky, práca so súbormi, prístup do databázy a podobne. [2]

PHP patrí medzi najrozšírenejšie programovacie jazyky pre webové aplikácie kvôli svojej jednoduchosti a bohatej zásobe funkcií. Poskytuje vysokú bezpečnosť pre aplikácie. Vyznačuje sa taktiež svojim výborným výkonom. Je open source, čím odpadajú rôzne licenčné poplatky a je okolo neho vytvorená obrovská komunita ľudí, ktorý vám vždy dokážu pomôcť a odpovedať na vaše otázky. [3]

#### **3.3.3 Apache**

Apache je softwarový webový server, ktorý zohrával kľúčovú úlohu v rannom rozmachu Internetu. V roku 2009 to bol prvý webový server, na ktorom bežalo viac než 100 miliónov webových stránok. Podporuje veľké množstvo funkcií. Väčšina z nich je implementovaná pomocou modulov,

---

ktoré rozširujú jadro. Môžu to byť funkcie podpory programovacích jazykov na strane serveru alebo rôzne autentizačné schémy. Podporované programovacie jazyky sú Perl, Python, Tcl a PHP. [4]

Webový server Apache bol zvolený hlavne kvôli svojej jednoduchosti a veľkej podpore funkcií. Ďalej odpadajú rôzne licenčné poplatky, pretože je open source. Beží najlepšie pod operačným systémom Linux, ktorý je taktiež open source. Napokon v neposlednom rade má obrovskú podporu zo strany vývojárov, ktorí ho stále vylepšujú a aktualizujú.

### **3.4 Klientská časť**

#### **3.4.1 Opis a funkcie**

Klientská časť beží na platforme Android. Táto časť zabezpečuje autorizáciu pomocou protokolu OAuth. Ďalej posiela požiadavky serverovej časti pomocou protokolu XML-RPC a získava od nej požadované dáta. Táto časť je napísaná v programovacom jazyku Java.

#### **3.4.2 Java**

Java je objektovo orientovaný programovací jazyk, ktorý vyvinula firma Sun Microsystems v roku 1995. Je to jeden z najpoužívanejších jazykov na svete. Napísaný a skompilovaný kód môže byť spustený na rôznych platformách. Aplikácie sú skompilované to takzvaného bytekódu, ktorý môže byť spustený na ľubovoľnom Java virtuálnom stroji bez ohľadu na architektúru počítača. Syntax je podobná programovacím jazykom C a C++. Tento programovací jazyk bol vyvíjaný aby splňoval 5 princípov:

1. Byť jednoduchý, objektovo orientovaný a známy.
2. Byť rozsiahly a bezpečný.
3. Byť nezávislý na architektúre a prenosný.
4. Dosahovať vysoký výkon.
5. Byť kompilovaný, dynamický a mal by podporovať prácu s vláknami.

[5]

Hlavným dôvodom prečo bola zvolená Java je, že klientská aplikácia je postavená na platforme Android (kapitola nižšie), ktorej aplikácie sú napísané primárne v tomto programovacom jazyku. Taktiež tento programovací jazyk je open source, tým pádom sa naňho nevzťahujú žiadne licenčné poplatky.

---

### 3.4.3 Android

Android je rozsiahla open source platforma, ktorá vznikla primárne pre mobilné zariadenia (inteligentné telefóny, tablety, navigácie, PDA). Zahrňuje v sebe operačný systém založený na jadre Linuxu. Bol vyvíjaný firmou Android Inc., ktorú v roku 2005 odkúpila spoločnosť Google. Architektúra operačného systému Android je rozdelená do 5 vrstiev. Každá vrstva má svoj účel a nemusí byť priamo oddelená od ostatných vrstiev. Tieto vrstvy sú (informácie čerpané z [6]):

**Jadro operačného systému** – tvorí abstraktnú vrstvu medzi používaným hardvérom a softvérom vo vyšších vrstvách. Jadro systému Android je postavené na Linuxe vo verzii 2.6 . Podporuje napríklad správu pamäti, správu sietí, správu procesov, multitasking.

**Knižnice** – využívajú rôzne komponenty systému. Sú napísané v programovacom jazyku C alebo C++.

**Android Runtime** – obsahuje aplikačný virtuálny stroj nazývaný Dalvik. Dalvik Virtual Machine (DVM) bol vytvorený primárne z 2 dôvodov. Prvým boli licenčné práva, kde jazyk Java a jeho knižnice sú voľné šíriteľné avšak Java Virtual Machine (JVM) nie. Ďalším dôvodom bola optimalizácia virtuálneho stroja pre mobilné zariadenia a to hlavne z hľadiska pomeru úspory energie a výkonu. V tejto vrstve sú takisto obsiahnuté základné knižnice programovacieho jazyku Java.

**Application Framework** – pre vývojárov najdôležitejšia. Poskytuje prístup k veľkému množstvu služieb, ktoré môžu byť použité priamo v aplikáciách.

**Základné aplikácie** – aplikácie, ktoré využívajú bežní užívatelia. Môže ísť o predinštalované aplikácie alebo aplikácie stiahnuté z Android Marketu. Príkladom je emailový klient, SMS program a podobne.

Táto platforma bola zvolená kvôli jej veľkému rozšíreniu a teší sa veľkej popularite. Keďže je open source, odpadá platba rôznych licenčných poplatkov.

### 3.4.4 Protokol XML – RPC

Tento protokol vykonáva vzdialené volanie procedúr (Remote Procedure Call - RPC). Na rozdiel od klasického RPC sú dáta zapuzdrené pomocou značkovacieho jazyka XML a prenášané protokolom HTTP. Táto koncepcia umožňuje aplikáciám, ktoré sú napísané v rôznych programovacích jazykoch vzájomnú komunikáciu. Komunikácia môže prebiehať medzi rôznymi počítačovými architektúrami alebo rôznymi operačnými systémami. Existuje veľké množstvo implementácií tohto protokolu pre rôzne programovacie jazyky ako napríklad Java, Python, C++, Perl atď. Tento protokol sa stal predlohou pre protokol SOAP. [7]



---

## 4 Sociálne siete

### 4.1 Čo je to sociálna sieť ?

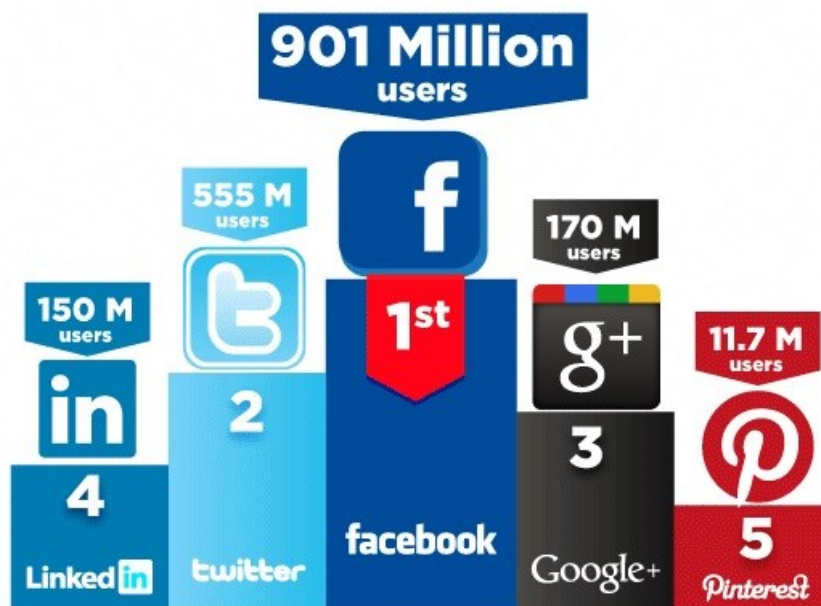
Sociálna sieť je prepojená skupina ľudí. V modernom a značne prevažujúcom ponímaní je sociálna sieť internetová služba, ktorá umožňuje registrovaným členom komunikovať medzi sebou a zdieľať rôzne informácie (fotky, videa, statusy). Niekedy sa pod týmto pojmom označujú aj diskusné fóra, kde si užívatelia vymieňajú názory a poznatky v rôznych témach. Komunikácia na sociálnych sieťach môže prebiehať buď medzi dvoma osobami alebo medzi osobou a celou skupinou užívateľov. Medzi najznámejšie sociálne siete súčasnosti patrí napríklad Facebook, Twitter, Google Plus, MySpace, LinkedIn. [8]

### 4.2 Ich prínos a ako ovplyvnili Internet

Sociálne siete môžeme považovať za takú malú revolúciu v oblasti Internetu. Úplne zmenili naše spôsoby komunikácie. V dnešnej dobe nemáme problém medzi sebou komunikovať na veľkú vzdialenosť (iný štát, iný svetadiel). Môžeme si písať s našim obľúbeným hercom/herečkou alebo inou populárnou osobnosťou. Môžeme byť členmi rôznych skupín, kde nájdeme ľudí s rovnakými názormi a koníčkami. Môžeme organizovať rôzne udalosti a zhromaždenia. Medzi ďalšie prínosy patrí napríklad, že si môžeme ľahšie nájsť novú prácu. Sociálne siete často využívajú ľudia, ktorý dohadzujú firmám nových zamestnancov. Na túto činnosť je práve určená sociálna sieť LinkedIn, kde môžete nahráť svoj životopis, projekty na ktorých ste pracovali, ukončené vzdelanie a podobne dôležité informácie na získanie zamestnania. V neposlednom rade môžu sociálne siete slúžiť aj ako marketingový nástroj, cez ktorý propagujete svoju firmu pomocou rôznych bannerov, reklám, ale aj aplikácií. Čo sa týka ovplyvnenia Internetu môžeme povedať, že Internet je teraz viac prepojený, otvorený a slobodnejší. Avšak všetko má aj svoje zápory. V poslednej dobe je hlavnou témou otázka súkromia užívateľov a zabránenie zneužitia týchto údajov na sociálnych sieťach proti samotným užívateľom. [9]

---

## 4.3 Príklady sociálnych sietí



Obrázok 4.1: Návštevnosť najpopulárnejších sociálnych sietí na jar v roku 2012 [10]

### 4.3.1 Facebook

Facebook je sociálna sieť, ktorá bola spustená v roku 2004 vlastnená firmou Facebook, Inc. V septembri 2012 dosiahla cez 1 miliardu používateľov a patrí medzi najväčšie a najznámejšie sociálne siete súčasnosti. Bola založená študentom Harvardskej univerzity Markom Zuckerbergom a jeho spolubývajúcimi Eduardom Saverinom a Andrewom McCollumom. Spočiatku členmi tejto sociálnej siete mohli byť iba študenti Harvardskej univerzity. Neskôr to mohli byť niektoré vybrané univerzity v USA. Napokon sa zakladatelia rozhodli sprístupniť túto sociálnu sieť celému svetu. [11]

Registrácia prebieha rýchlo a bez problémov. Jedinou podmienkou je vek nad 13 rokov. Zvolíte si emailovú adresu, ktorú budete používať pri prihlásení, vyplníte vaše meno, priezvisko a heslo. Napokon musíte ešte odsúhlasiť licenčné podmienky. Po registrácii je účet okamžite aktívny a môžete sa prihlásiť.

Každý užívateľ má v profile svoju Timeline, na ktorú mu môžu ostatní užívatelia písať odkazy a zobrazujú sa tu všetky udalosti, ktoré sa dejú na sociálnej sieti zoradené podľa mesiaca v roku, v ktorom sa uskutočnili. Užívateľ môže zdieľať taktiež rôzne fotografie alebo videá. Príspevky môžete vy alebo ostatní užívatelia komentovať, označiť symbolom Páči sa mi to, prípadne zdieľať na svojej ploche Timeline.

Priateľstvá tu fungujú na princípe takzvanej obojsmernej relácie. To znamená, že keď si chceme niekoho pridať medzi svojich priateľov pošleme mu žiadosť, ktorú dotyčný musí akceptovať.

---

Ak ju akceptuje, zobrazí sa nám v zozname našich priateľov a taktiež jemu sa zobrazí náš profil medzi priateľmi. V prípade odmietnutia zostanú zoznamy priateľov oboch ľudí nezmenené. Priateľov si môžeme triediť do rôznych skupín, ktoré si môžeme sami definovať (rodina, práca, škola a podobne).

Užívatelia majú možnosť na tejto sociálnej sieti vytvárať, administrovať a pridávať sa do skupín. Tieto skupiny môžu mať rôzne účely. Združovať ľudí z rovnakej firmy, školy, s rovnakými záujmami a podobne. Skupiny môžu byť verejné alebo súkromné. Do verejných skupín sa môže pridať ktorýkoľvek užívateľ na sociálnej sieti. Čo sa týka súkromných skupín, tam sa užívateľ môže dostať len cez pozvánku, ktorú dostane od nejakého existujúceho člena alebo administrátora skupiny.

Facebook ďalej podporuje tvorbu rôznych udalostí. Sú nimi rôzne školské stretnutia, firemné akcie alebo výstavy, prípadne aj protesty alebo petície. Pri vytváraní udalosti užívateľ vyplní jej dátum, čas, miesto konania, krátky popis a napokon či bude táto udalosť verejná (ktokoľvek sa môže zúčastniť) alebo súkromná (iba pozvaní ľudia sa môžu zúčastniť). Pokiaľ je administrátorom tejto udalosti, má právo kedykoľvek pridávať alebo odoberať pozvaných respektíve zúčastnených ľudí.

Facebook prináša v celku obsiahle nastavenia súkromia. Do týchto nastavení môžeme zaradiť napríklad nastavenie viditeľnosti profilu, skrývanie určitých informácií, statusov, fotiek, či sa má náš profil zobrazovať vo vyhľadávačoch ako je napríklad Google a podobne. Implicitne máme definované 3 druhy zobrazovania údajov. Verejné, kde každý užívateľ na sociálnej sieti alebo mimo nej vidí naše informácie. Druhé v poradí je nastavenie priateľa, kde vaše informácie vidia len Vaši priatelia. Tretie je priatelia mojich priateľov. Napokon je možnosť definovať vlastné a to tým spôsobom, že si dotýčnú osobu alebo skupinu priateľov vyhľadáme podľa mena a povolíme/zakážeme vidieť naše informácie.

#### 4.3.2 Google Plus

Google Plus sa radí taktiež medzi najpopulárnejšie sociálne siete súčasnosti. Vlastní ju firma Google, ktorá je známa predovšetkým vďaka svojmu internetovému vyhľadávaču. Táto sociálna sieť bola spustená v lete roku 2011. V decembri 2012 mala vyše 500 miliónov registrovaných užívateľov. Na rozdiel od ostatných sietí, ktoré tvorí jedna stránka Google Plus sa dá považovať ako jedna vrstva (nazývaná aj sociálna vrstva) v sade nástrojov, ktoré získate po vytvorení vášho Google účtu. [12]

Registrácia je veľmi jednoduchá. V nej zadáte svoje meno, priezvisko, login z ktorého sa vám následne vytvorí nová emailová schránka na službe Gmail a heslo. Odsúhlasíte licenčné podmienky a službu môžete okamžite začať využívať. S registráciou získate okrem vyššie spomínanej emailovej schránky rôzne služby od Googlu (Google Calendar, Google Drive, Google Docs, atď.), kde sa budete prihlasovať pod svojim názvom tejto emailovej schránky a príslušným heslom.

---

Na úvodnej stránke máte zobrazené najnovšie príspevky (v Google Plus nazývané aktivity), ktoré publikovali vaši priatelia alebo skupiny, ktorých ste členom. Podobne ako na Facebooku môžete tieto aktivity komentovať, zdieľať alebo ich označiť symbolom +1, ktorý má rovnaký význam ako symbol „Páči sa mi to“ na sociálnej sieti Facebook.

Ďalším významným pojmom na Google Plus sú kruhy. Plnia úlohu skupín, do ktorých si ukladáte priateľov. Niektoré skupiny máte vytvorené už pri prvej návšteve (blízki priatelia, rodina, práca). Na Google Plus nie je povinná obojsmerná relácia medzi priateľmi ako napríklad na Facebooku. To znamená, že keď si chcete niekoho pridať medzi svojich priateľov, tak dotýčny musí túto žiadosť schváliť a pridať si taktiež vás medzi svojich priateľov. Na Google Plus príde dotýčnej osobe oznámenie na email, že ste si ju pridali medzi priateľov a je len na nej, či si pridá aj ona vás. Samozrejme, že možnosti komunikácie sú rozsiahlejšie v situácii, keď platí obojsmerná relácia.

Komunity na Google Plus označujú stránky, ktoré fungujú podobne ako skupiny na Facebooku. Nájdete v nich ľudí s rovnakými záľubami alebo názormi. Máte na výber zo 4 možností nastavení. 2 druhy verejných a 2 druhy súkromných. V prípade verejného nastavenia sú tieto komunity indexované a dajú sa vyhľadať pomocou vyhľadávača Google. Prvá možnosť verejného nastavenia zahŕňa, že každý užívateľ na sociálnej sieti sa môže pridať do tejto komunity a vytvárať, komentovať alebo zdieľať jej príspevky. Druhá možnosť je podobná s tým rozdielom, že vytvárať príspevky môžu len administrátori dotýčnej komunity avšak komentovať a zdieľať ich môžu aj členovia. Súkromné nastavenie zahŕňa, že sa do komunity môžu pridať len užívatelia, ktorí dostali pozvánku. [13]

Udalosti na Google Plus majú rovnakú funkciu ako na sociálnej sieti Facebook. Umožňujú ľuďom vytvárať rôzne stretnutia so svojimi priateľmi alebo kolegami z práce. Majú však aj svoje výhody oproti Facebooku. Pri tvorbe udalosti si môžete zvoliť jeden z motívov, ktorý sa hodí k dôvodu udalosti (svadba, narodeniny, grilovanie a pod.). Udalosti sa automaticky pridávajú do služby Google Calendar, kde si užívateľ ukladá všetky dôležité dátumy. V dobe udalosti máte možnosť zdieľať fotky v reálnom čase a všetky ich máte zoskupené na jednom mieste. Napokon je možný videochat s pozvanými ľuďmi za účelom dohodnutia sa na mieste stretnutia, prípadne oznámení o zmene dátumu a času konania. [14]

Google Plus ponúka veľké možnosti nastavenia súkromia pre svojich užívateľov. Na začiatok treba porozumieť konceptu kruhov. Každý kruh (skupina ľudí) sa odlišuje. Kým ľudia, ktorých máte uložených v blízkych priateľoch vidia všetky vaše príspevky, ľudia nachádzajúci sa v rodinných príslušníkoch vidia predovšetkým príspevky a fotky týkajúce sa vašej rodiny. Pri definovaní vlastného kruhu si môžete tieto nastavenia modifikovať. Ďalej si môžete zmeniť viditeľnosť kruhov voči

---

ostatným užívateľom sociálnej siete. Záleží len na vás či vaše kruhy budú môcť prezerat' len Vaši priatelia alebo všetci. Ďalej máte možnosť zakázať vyhľadávateľom zobrazovať váš profil (špeciálne vo vyhľadávateľi Google). Napokon stojí za zmienku možnosť určiť pri písaní príspevku alebo zdieľaní fotografie, ktorým kruhom sa táto aktivita zobrazí. [15]

#### 4.3.3 Twitter

Twitter je sociálna sieť, ale taktiež sa o nej dá hovoriť ako o službe mikrobloggeru. Bola založená v roku 2006 Jackom Dorsey a odvtedy získala veľký význam a popularitu na celom svete. Teraz ju vlastní firma Twitter Inc. V roku 2012 využívalo túto sociálnu sieť 500 miliónov užívateľov, pričom každý deň sa pošle asi 340 miliónov tweetov (príspevkov). Twitter je 10. najčastejšie navštevovaná stránka Internetu a býva označovaná aj ako SMS Internetu. [16]

Registrácia prebieha veľmi jednoducho podobne ako na vyššie spomenutých sociálnych sieťach. Vyplníte jeden formulár, kde zadáte svoje celé meno, prihlasovacie meno, email a heslo. Odsúhlasíte licenčné podmienky a môžete okamžite využívať svoj vytvorený účet.

Na rozdiel od sociálnych sietí ako Facebook alebo Google Plus, kde ľudia píšu rozsiahle príspevky v službe Twitter je správa obmedzená na 140 znakov. Príspevky sa tu nazývajú tweety. Má to samozrejme svoje výhody, ale aj nevýhody. Výhodou je fakt, že ľudia píšu len naozaj dôležité informácie a nemáte sa problém zorientovať vo veľkom množstve príspevkov. Na druhej strane je zas niekedy potrebné podať detailnejšie informácie a v tomto Twitter zlyháva. Kvôli tomuto obmedzeniu na 140 znakov sa veľmi často používajú skrátene URL odkazy a špeciálne znaky ako @ alebo #.

Znak # (nazývaný hashtag) pre určitým slovom slúži na kategorizáciu vašich tweetov a umožňuje ich rýchlejšie vyhľadať. Znak @ pre určitým slovom sa používa ako označenie profilu osoby, ktorej odpovedáme na príspevok alebo ju len chceme spomenúť v našom príspevku. [17][18]

Nachádzajú sa tu dve skupiny užívateľov (priateľov). V jednej skupine sú ľudia, ktorých „sledujete“ (odoberáte príspevky) a v druhej naopak ľudia, ktorí „sledujú“ (odoberajú príspevky) od vás. Nato aby ste od niekoho mohli odoberať príspevky nepotrebuje súhlas dotyčného užívateľa. Taktiež ani ony od vás nepotrebujú súhlas. Avšak zakaždým, keď niekto začne odoberať vaše príspevky, služba vás nato upozorní pomocou emailu. Máte možnosť si vytvárať zoznamy, do ktorých môžete ukladať a triediť ľudí.

Vaše tweety si môžete ochrániť tým, že špecifikujete ktorí užívatelia ich môžu odoberať. Taktiež Twitter sleduje, z ktorých stránok odoberáte príspevky a podľa toho Vám ponúka ďalšie stránky, ktoré by sa Vám mohli páčiť. Túto vlastnosť možno tiež vypnúť v nastaveniach za účelom zvýšenia súkromia. [19]

---

#### 4.3.4 LinkedIn

LinkedIn je profesionálne zameraná sociálna sieť, na ktorej sa stretávajú väčšinou ľudia ohľadom práce a pracovných záujmov. Bola spustená 5. mája v roku 2003 určená primárne pre profesionálov. V januári roku 2013 štatistiky ukazovali viac než 200 miliónov užívateľov z viac než 200 krajín sveta. [20]

Registrácia prebieha veľmi jednoducho ako na ostatných populárnych sociálnych sieťach. Stačí vyplniť svoje meno, email a heslo. Následne odsúhlasiť licenčné podmienky a účet máte okamžite aktivovaný.

Na tejto sociálnej sieti nebudete vidieť veľa rozsiahlych statusov alebo blogov. Ľudia tu väčšinou iba posielajú odporúčania na iných ľudí, prezerajú si váš profil a až potom vám napíšu nejakú tú súkromnú správu.

Skupiny majú omnoho väčší význam. Môžete sa pripájať do rôznych záujmových skupín s rovnakými pracovnými zručnosťami alebo dokonca do skupín malých či veľkých firiem. Skupiny firiem ponúkajú niekedy aj zoznam voľných pracovných pozícií a požiadavky na ne. Avšak veľkou nevýhodou je, že v týchto skupinách sa vyskytuje veľké množstvo spamu, pretože tu nie je implementovaný anti - spamový filter. Niektoré spoločnosti na tejto sociálnej sieti ponúkajú práve takéto služby na zabezpečenie firemných skupín. [21]

Medzi dvoma užívateľmi tu funguje takzvaná obojsmerná relácia ako napríklad na Facebooku. To znamená, že keď si niekoho chcem uložiť do svojich priateľov najprv mu musím poslať žiadosť, ktorú dotyčný potvrdí (tým pádom si aj on vás uloží medzi svojich priateľov). Svojich priateľov si môžete ukladať a triediť do skupín podľa rôznych kritérií (blízki priatelia, kolegovia, podľa názvu firmy, podľa pracovnej pozície atď.). Ďalej ako som už vyššie spomenul môžete svojich priateľov odporučiť ostatným užívateľom alebo konkrétnym firmám.

Napriek tomu, že LinkedIn je profesionálna sociálna sieť kde nezdiate svoje fotky z party alebo príspevky, ktoré nemusí každý pochopiť v dobrom aj tu si musíte strážiť svoje súkromie a nedovoliť každému prezerať vaše súkromné informácie. Pretože predvolebne majú všetci ľudia prístup k vášmu profilu bez ohľadu nato či sú zaregistrovaní alebo nie. Môžu vidieť váš životopis, kde pracujete, na akej pozícii ste, predošlé zamestnania a podobne. Ďalej môžete zmeniť kto môže vidieť vaše novinky, ktoré ste dostali od ostatných, či už ide o odporúčania alebo prístup do nových skupín. Toto všetko sa zobrazuje vo vašom profile. LinkedIn je taktiež prepojený so sociálnou sieťou Twitter. Môžete ich medzi sebou prepojiť a keď napíšete nejaký príspevok na jednej, príspevok sa zobrazí taktiež na druhej sociálnej sieti. Ďalším nastavením je viditeľnosť častí vášho profilu a taktiež vidíte

---

kto videl váš profil. Napokon máte možnosť spravovať rôzne aplikácie (ako na Facebooku), ktoré prístupujú k Vaším údajom z profilu. [22]

---

## 5 Prepojenie sociálnych sietí a aplikácií

### 5.1 Dôvody

V dnešnej dobe sociálne siete patria medzi najviac rozšírené služby na Internete. Veľmi populárne je prepojenie iných webových stránok, služieb alebo aplikácií so sociálnymi sieťami. Dôvodom tohto prepojenia je najčastejšie použitie prihlasovacích údajov, ktorými sa užívateľ pripája na sociálnu sieť. Pri registrovaní napríklad na novej stránke si užívateľ nemusí vytvárať nový login a heslo. Ďalšie využitie môže byť jednoduchá komunikácia s konkrétnou sociálnou sieťou. Vtedy môže užívateľ pomocou klientskej aplikácie pristupovať napríklad k svojmu zoznamu priateľov, písať alebo komentovať príspevky a podobne. Napokon dôvodom môže byť prepojenie viacerých sociálnych sietí do jednej aplikácie. Keď užívateľ napríklad napíše status na Facebooku automaticky sa tento status skopíruje a pošle na Twitter a LinkedIn. K prepojeniu aplikácií so sociálnymi sieťami sa využíva takzvaného API a autorizačných protokolov OpenID a OAuth.

### 5.2 API

Táto skratka znamená Application Programming Interface. Jedná sa o zbierku procedúr, funkcií, tried alebo protokolov nejakej knižnice, ktorú môže programátor využívať. API určuje akým spôsobom sú funkcie volané zo zdrojového kódu programu. Väčšinou sa používa ako prepojenie medzi dvoma komunikujúcimi softwarovými komponentmi.

Pre lepšiu predstavu si môžeme uviesť príklad. Máme napríklad nejakú webovú stránku veľkej siete reštaurácií. Chceme našim zákazníkom sprístupniť zoznam miest našich reštaurácií pomocou služby Google Maps. Toho docielime pomocou Google API, ktorá sa postará o komunikáciu medzi touto službou a našou webovou stránkou. Webová stránka bude posilať požiadavky cez toto API a taktiež bude cez toto API prijímať aj odpovede z Google serveru. [23]

V dnešnej dobe všetky najpopulárnejšie sociálne siete majú svoje API, pomocou ktorého môžete prepojiť danú sociálnu sieť s vašou aplikáciou. Medzi tieto sociálne siete patrí napríklad Facebook, Twitter, Google Plus, LinkedIn atď.

### 5.3 Protokol OAuth

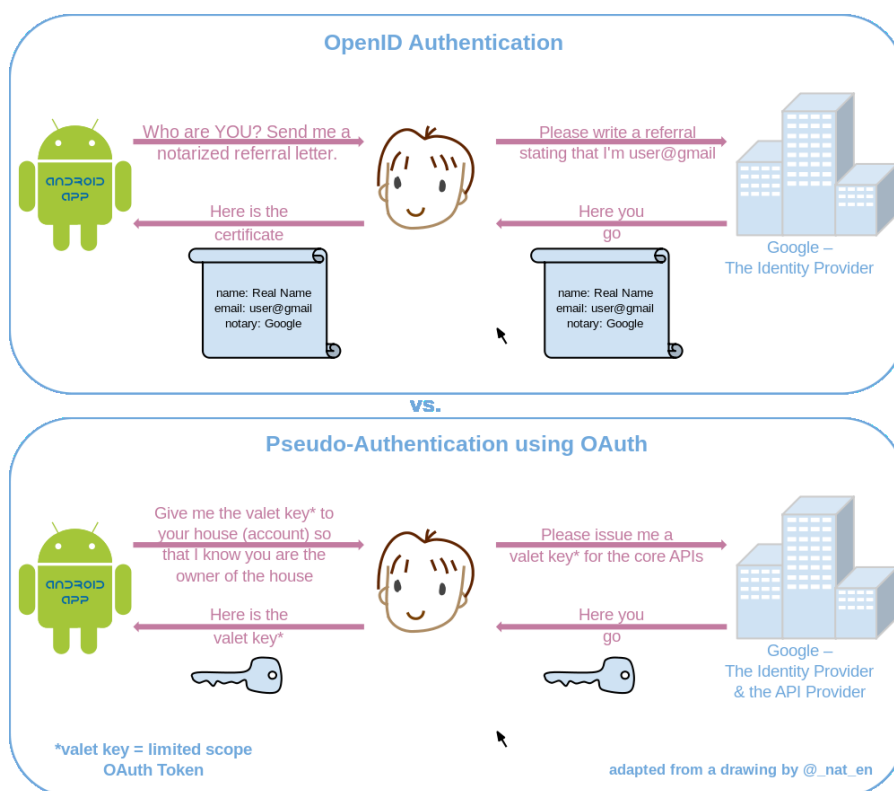
Tento protokol je otvorený štandard pre autorizáciu. Poskytuje koncovým užívateľom prístup k serveru bez toho aby museli zdieľať svoje prihlasovacie údaje s daným serverom. Tento protokol je odlišný od protokolu OpenID. Používa sa hlavne vo verziách 1.0 a 2.0. Tieto dve verzie nie sú navzájom kompatibilné.



Autorizácia pomocou protokolu OpenID prebieha nasledovne. Aplikácia si od klienta vyžiada identitu klienta, ktorá prebieha väčšinou prihlásením klienta pomocou svojho unikátneho OpenID URI namiesto klasických prihlasovacích údajov. Aplikácia sa pri autorizovaní spolieha na dôveryhodnosť poskytovateľa OpenID.

Princíp autorizácie pomocou protokolu OAuth je v získaní dočasného prístupového tokenu, ktorý zašle aplikácia pomocou API klientovi. S týmto prístupovým tokenom môže klient komunikovať pomocou API s aplikáciou. Samotný protokol nevyžaduje zadávanie prihlasovacích údajov od klienta. Avšak môže sa stať, že aplikácia po autorizovaní bude chcieť prideliť klientovi unikátnu identitu pomocou, ktorej bude identifikovateľný. Vtedy môže aplikácia požadovať prihlásenie a zadanie prihlasovacích údajov. V tomto prípade na rozdiel od protokolu OpenID sa aplikácia spolieha na dôveryhodnosť svojho vlastného API, nie na dôveryhodnosť poskytovateľa OpenID identity, čo znamená vyššiu záruku bezpečnosti. [24]

Vyššie dva spomenuté postupy popisuje obrázok 5.1



Obrázok 5.1: Autorizácia pomocou OpenID a OAuth [25]

---

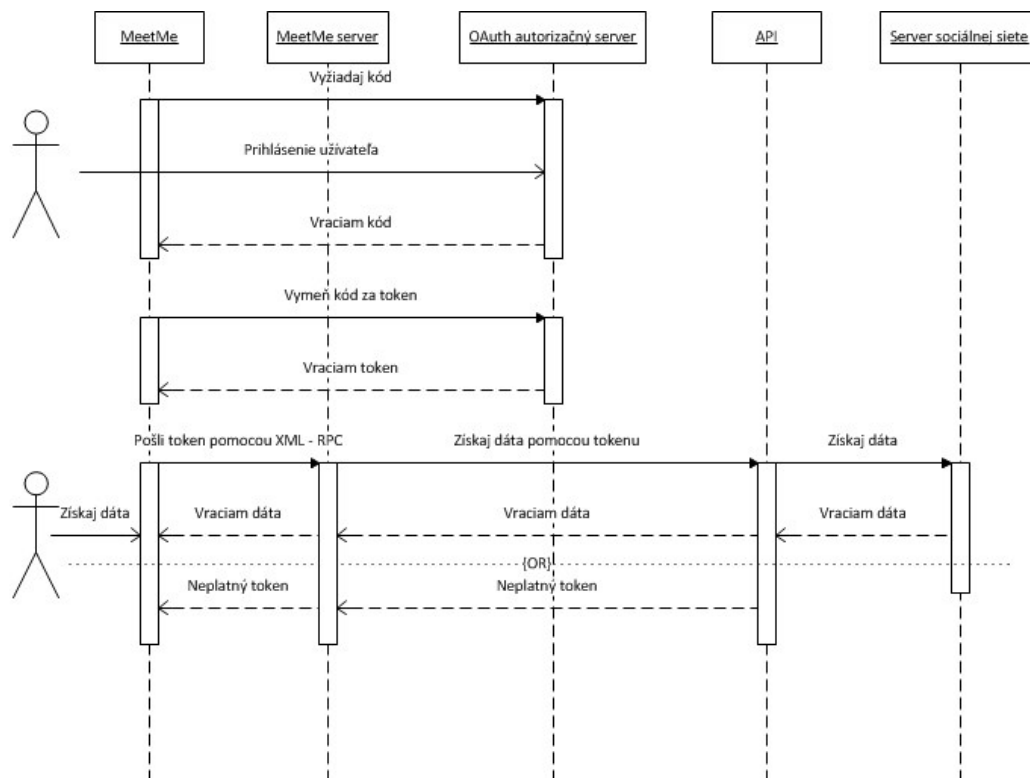
## 5.4 Prepojenie s MeetMe

Primárna funkcia aplikácie MeetMe je prepojenie rôznych sociálnych sietí. Preto môžeme očakávať použitie protokolu OAuth ako aj komunikáciu s rôznymi API pre danú sociálnu sieť. Každá sociálna sieť, s ktorou aplikácia MeetMe bude komunikovať má vlastnú API pre rôzne programovacie jazyky. Na internete môžeme nájsť API napríklad pre PHP, Javu, Objective C, Javascript atď. Každá API používa vlastné sady knižníc a funkcií, ktoré zabezpečujú autorizáciu pomocou protokolu OAuth a následné získanie prístupového tokenu. Ďalej zabezpečujú posielanie autorizovaných požiadaviek na server sociálnej siete a získané dáta z tejto sociálnej siete posielajú naspäť klientskej aplikácii.

---

## 6 Implementácia konektorov do aplikácie MeetMe

### 6.1 Pribeh komunikácie medzi MeetMe a sociálnymi sieťami

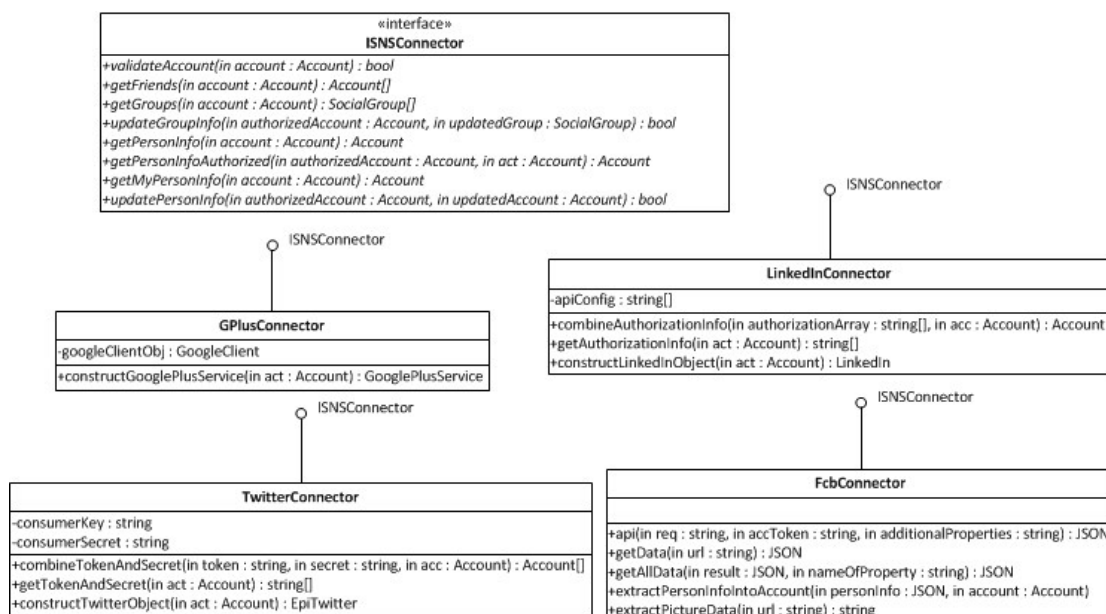


Obrázok 6.1: Popis komunikácie medzi MeetMe a sociálnymi sieťami

Obrázok 6.1 popisuje spôsob ako prebieha získanie prístupového tokenu a získavanie dát zo sociálnej siete pomocou tohto prideleného tokenu. Tento proces je vo svojej podstate rovnaký pri všetkých sociálnych sieťach, ktoré implementuje aplikácia MeetMe. Jednotlivé odchýlky sú popísané v nižších kapitolách, kde sa rozoberá každý konektor zvlášť. Jednu vec majú však tieto sociálne siete spoločnú. Pred samotným použitím konkrétnej API si musíte vašu aplikáciu najprv zaregistrovať na stránkach danej sociálnej siete. Postup ako to urobiť nájdete v kapitole 9. Prílohy.

## 6.2 Opis dôležitých častí na strane serveru

### 6.2.1 Rozhranie ISNSConnector



Obrázok 6.2: UML diagram, ktorý zobrazuje prepojenie konektorov s rozhraním

Každý konektor musí implementovať rozhranie ISNSConnector, v ktorom sú definované metódy potrebné na jeho správnu funkčnosť. Sú to tieto metódy:

**validateAccount** – metóda, pomocou ktorej sa zisťuje či objekt typu Account zadany ako vstupný parameter sa zhoduje s objektom typu Account, ktorý je práve prihlásený v klientskej aplikácii. Porovnávajú sa hodnoty premennej serviceId oboch objektov typu Account. Ak sú tieto hodnoty rovnaké, jedná sa o toho istého užívateľa. V opačnom prípade nie.

**getFriends** – vracia pole objektov typu Account, ktoré reprezentujú profily všetkých priateľov, ktorých má prihlásený užívateľ na danej sociálnej sieti.

**getGroups** – vracia pole objektov typu SocialGroup. Každý takýto objekt reprezentuje jednu skupinu, ktorú má prihlásený užívateľ vytvorenú na danej sociálnej sieti. Ďalej každý takýto objekt obsahuje pole objektov typu Contact, kde každý objekt typu Contact uchováva ID člena danej skupiny.

**updateGroupInfo** – používa sa na zmenu určitého parametra objektu typu SocialGroup. Požaduje dva vstupné parametre. Jeden objekt typu Account a jeden objekt typu SocialGroup. Objekt typu Account musí obsahovať prístupový token a objekt typu SocialGroup určuje, ktorá skupina užívateľov sa bude upravovať.

**getPersonInfo** – vracia objekt typu Account, ktorý obsahuje dáta o určitom užívateľovi nachádzajúcom sa na danej sociálnej sieti. Vstupným parametrom je taktiež objekt typu Account, v

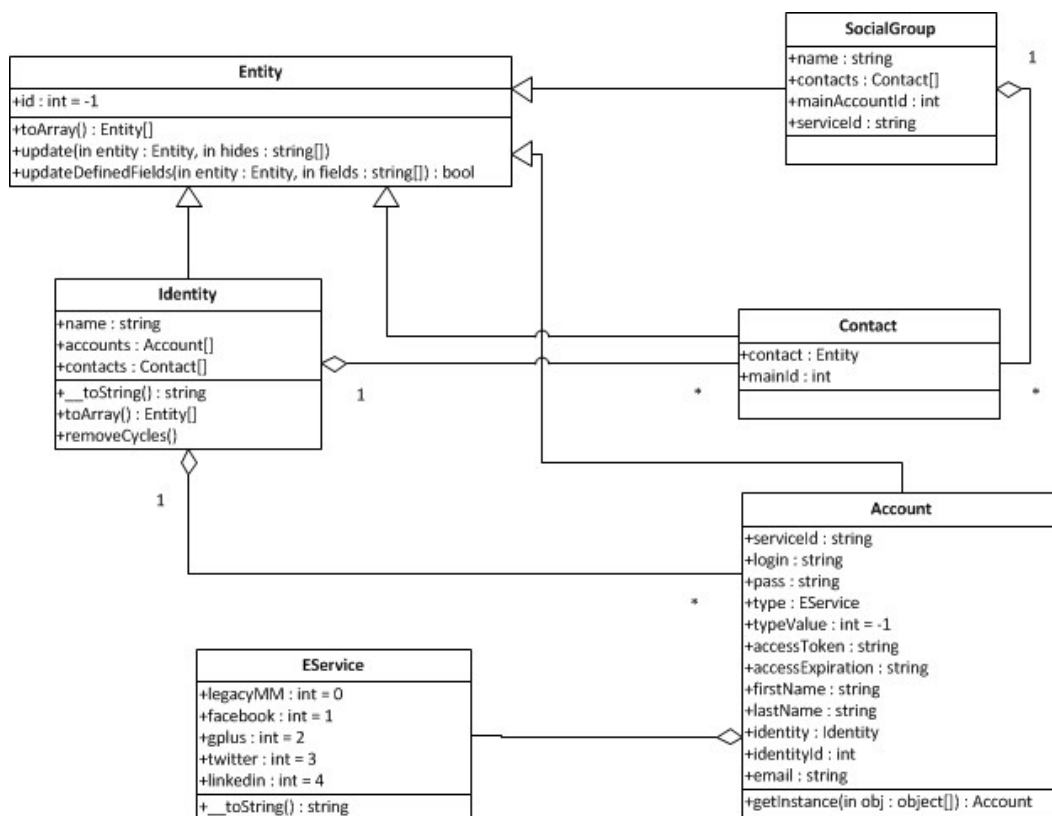
ktorom je uložené ID užívateľa, ktorého informácie chceme získať a prístupový token. Pomocou tohto tokenu je možné vytvoriť autorizovanú požiadavku na autorizačný server.

**getPersonInfoOAuthorized** - vracia objekt typu Account, ktorý obsahuje dáta o určitom užívateľovi nachádzajúcom sa na danej sociálnej sieti. Na rozdiel od vyššie spomínanej metódy getPersonInfo táto metóda požaduje dva parametre. Obidva sú objekty typu Account. Prvý určuje kto sa má vyhľadať (podľa jeho serviceId) a druhý Account objekt obsahuje prístupový token, ktorý sa vloží do hlavičky autorizovaného požiadavku.

**getMyPersonInfo** – vracia objekt typu Account, ktorý obsahuje dáta o aktuálne prihlásenom používateľovi. Ako vstupný parameter požaduje objekt typu Account, v ktorom je uložený prístupový token.

**updatePersonInfo** - používa sa na zmenu určitého parametra objektu typu Account. Vstupné parametre sú dva objekty typu Account. Prvý z nich musí obsahovať prístupový token a druhý objekt typu Account určuje, ktorý účet sa bude upravovať.

## 6.2.2 Dôležité objekty, s ktorými pracujú konektory



Obrázok 6.3: UML diagram, ktorý zobrazuje vzťahy medzi dôležitými objektmi

---

**Account** – tento objekt udržiava všetky potrebné informácie o jednom užívateľskom účte. Medzi tieto dôležité informácie patrí napríklad unikátny identifikátor, prístupový token, expirácia tohto prístupového tokenu, meno, email atď.

**EService** – objekt, ktorý určuje akú sociálnu sieť objekt typu Account používa.

**Contact** – na rozdiel od objektu typu Account tento objekt uchováva iba unikátny identifikátor užívateľa sociálnej siete (priateľa).

**SocialGroup** – tento objekt predstavuje jednu skupinu priateľov užívateľského účtu. Objekt uchováva identifikátor, názov skupiny a pole objektov typu Contact.

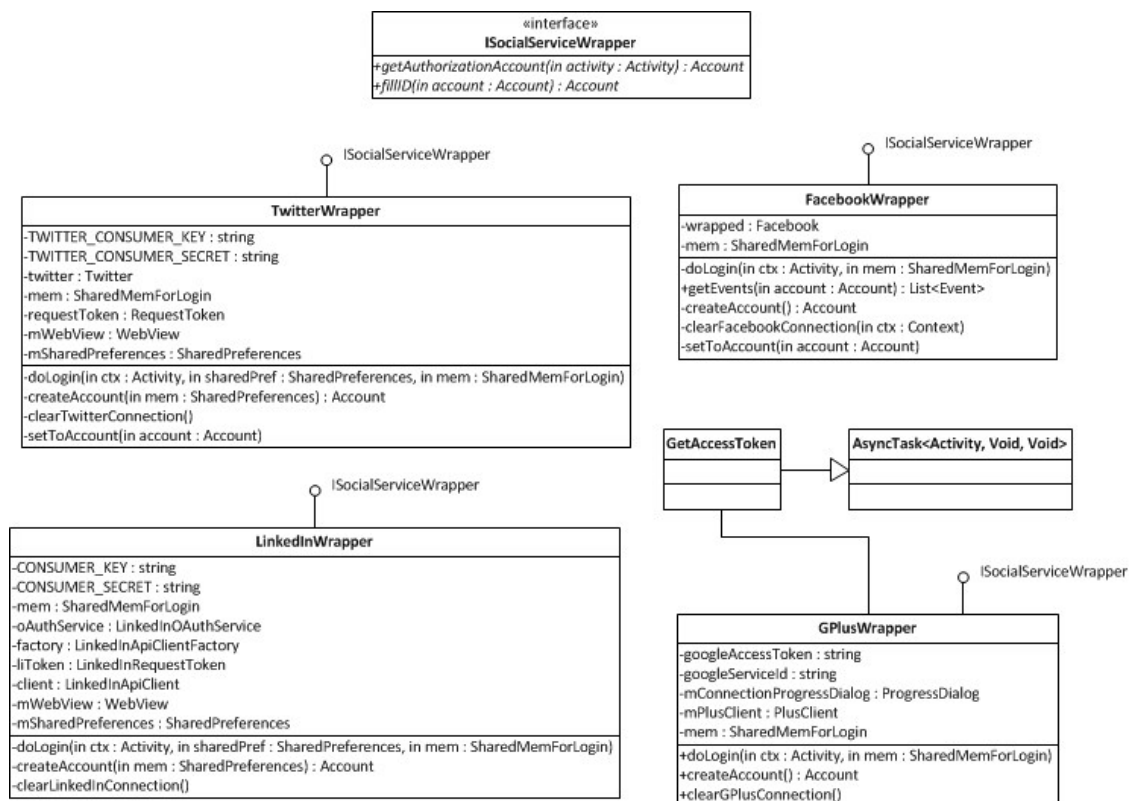
**SNSConfiguration** - v tomto objekte sú uložené všetky dôležité konštanty potrebné na prepojenie servera aplikácie MeetMe a PHP API danej sociálnej siete. Príkladom sú premenné ako twittConsumerKey, gPlusClientKey, linkedInAppSecret atď.

#### 6.2.3 Súbory `twitter_functions.php`, `gplus_functions.php` a `linkedin_functions.php`

V týchto súboroch sú uložené podporné funkcie, ktoré komunikujú s PHP API knižnicami.

## 6.3 Opis dôležitých častí na strane klienta

### 6.3.1 Rozhranie ISocialServiceWrapper



Obrázok 6.4: UML diagram, ktorý zobrazuje prepojenie wrapperov s rozhraním

Každý wrapper v Android aplikácii implementuje toto rozhranie, v ktorom sú definované dve kľúčové metódy.

**getAuthorizationAccount** – metóda vracia objektu typu Account. Jej funkciou je spustiť autorizačný proces a po úspešnej autorizácii vytvoriť nový objekt typu Account, v ktorom je uložený prístupový token. Ako vstupný parameter vyžaduje objekt typu Activity a môže vyhadzovať výnimku `NotConnectedException`.

**fillID** – táto metóda je zavolaná po úspešnej autorizácii a vytvorenia nového objektu typu Account. Má na starosť získanie unikátneho identifikátoru pre objekt typu Account. Ako vstupný parameter požaduje jeden takýto objekt, v ktorom je už uložený prístupový token. Metóda vracia objekt typu Account, v ktorom je okrem prístupového tokenu uložený aj identifikátor

### 6.3.2 Dôležité objekty, s ktorými pracujú wrappery

**Account** – tento objekt udržiava všetky potrebné informácie o jednom užívateľskom účte. Medzi tieto dôležité informácie patrí napríklad unikátny identifikátor, prístupový token, expirácia tohto prístupového tokenu, meno, email atď.

---

**EService** – objekt, ktorý určuje akú sociálnu sieť objekt typu Account používa.

**SocialServiceWrapperProvider** – objekt, ktorý sa stará o vytváranie a správu jednotlivých wrapperov. Implementuje návrhový vzor Factory.

**Configuration** – tento objekt uchováva všetky nastavenia ohľadom pripojenia sa na PHP server, s ktorým komunikuje aplikácia MeetMe. V komunikácii medzi aplikáciou a serverom sa využíva protokol XML-RPC.

## 6.4 Google Plus konektor a wrapper

### 6.4.1 Princíp fungovania

Pri tvorbe aplikácií, ktoré komunikujú so službami Google Plus sa používa Google Plus API. Je to zbierka knižníc, ktoré majú na starosti autentifikáciu a autorizáciu pomocou Google konta a následne zabezpečujú prenos informácií medzi Google Plus a danou aplikáciou.

Proces autorizácie prebieha nasledovne. Aplikácia si vyžiada pre jej užívateľa takzvaný prístupový token od autorizačného serveru Google. Užívateľ je presmerovaný na prihlasovací formulár stránky Google, kde vyplní svoju emailovú adresu v službe Google a k nej príslušné heslo. V prípade, že tieto prihlasovacie údaje sú nesprávne server vyzve užívateľa zadať tieto údaje znovu. Ak boli tieto prihlasovacie údaje zadané správne server pošle odpoveď aplikácii, ktorá dostane špeciálny kód pre daného užívateľa pridelený službou. Tento kód pošle aplikácia autorizačnému serveru v ďalšej požiadavke. V prípade, že tento kód je správny, server pridelí danému užívateľovi unikátny prístupový token, ktorý následne pošle aplikácii vo svojej odpovedi. Tá môže túto hodnotu uložiť napríklad do SESSION premennej. Ďalej je už komunikácia jednoduchá. Pri každej požiadavke na službu Google Plus sa pridá hlavička Authorization, ktorá obsahuje daný prístupový token. V prípade, že operácie s hlavičkami nie sú dostupné prístupový token sa môže poslať serveru Google aj ako GET parameter v URL adrese. Ak token súhlasí s tým, ktorý služba pridelila užívateľovi aplikácie server Google v odpovedi odošle požadované informácie. V opačnom prípade odošle chybu o zlom prístupovom tokene. [26]

### 6.4.2 Google Plus API

Pri implementácii tohto konektoru som použil voľne dostupný Google PHP API klient, ktorý je schopný pracovať z rôznymi službami od Google ako sú napríklad Google Maps, Google Contacts, Google Plus a podobne. Tento klient používa na prácu so službou Google Plus dva objekty. Jeden sa nazýva Google\_Client a druhý Google\_PlusService.

**Google\_Client** - ukladá nastavenia z Google API Console, ktoré sú špecifické pre Vašu aplikáciu a zabezpečuje získanie prístupového tokenu pre konkrétneho užívateľa.



---

**Google\_PlusService** – vytvára sa pomocou vyššie spomínaného objektu. V konštruktore vyžaduje **Google\_Client** objekt. Zabezpečuje komunikáciu medzi aplikáciou a službou Google Plus a posíla jej autorizované požiadavky kde do hlavičky **Authorization** vkladá prístupový token, ktorý užívateľ získal.

#### 6.4.3 Funkcie v konektore

**Google\_Client** objekt sa vyžaduje v konštruktore konektoru. Tento **Google\_Client** objekt sa následne uloží do súkromnej premennej **googleClientObj**, pretože tieto údaje sa v priebehu komunikácie už meniť nebudú. Pri volaní ostatných metód ako napríklad **validateAccount**, **getPersonInfo** a podobne sa vytvorí na začiatku objekt **Google\_PlusService** a uloží sa do lokálnej premennej. Uložením do lokálnej premennej sa zachováva reentrantnosť konektoru. Tvorba tohto objektu prebieha jednoducho. Vytvorí sa nový **Google\_PlusService** objekt pomocou uloženého **Google\_Client** objektu v súkromnej premennej konektoru **googleClientObj**. Následne sa upraví prístupový token tohto objektu podľa prístupového tokenu **Account** objektu, ktorý je vstupným parametrom tejto metódy. Potom pomocou tejto lokálnej premennej sa posielajú požiadavky službe Google Plus. Získané dáta sú uložené vo formáte JSON. Z týchto dát sa vyberú tie, ktoré potrebujeme a do novo vytvoreného **Account** objektu (prípadne poľa objektov tohto typu) ich uložíme. Nakoniec ich pošleme klientskej aplikácii.

#### 6.4.4 Funkcie vo wrapperi

Princíp autorizácie je v Google Plus wrapperi trochu iný ako v ostatných wrapperoch. Využíva sa tu služba Google Play Services, ktorú musíte mať nainštalovanú v telefóne. Pomocou tejto služby sa pri autorizačnom procese použije Google účet nastavený v telefóne. Tým pádom odpadá vyplňovanie nejakého prihlasovacieho formulára pri získavaní prístupového tokenu. Okrem toho, že túto službu musíte mať nainštalovanú musí váš telefón bežať minimálne na Androide verzii 2.2 . Na začiatku wrappera sú deklarované potrebné objekty na prácu s Google Android API. Metóda, ktorá sa stará o celý autorizačný proces je metóda **doLogin**. V nej sa vytvorí objekt typu **PlusClient**, spolu s potrebnými rozhraniami, v ktorých sa definujú rôzne akcie na základe stavov **PlusClient** objektu. Musíte definovať čo sa má spraviť po úspešnom prihlásení alebo odhlásení a pri výskyte nejakej chyby počas autorizačného procesu. Po úspešnom prihlásení sa vytvára objekt typu **GetAccessToken**., Tento objekt sa dedí z objektu **AsyncTask**. Objekt typu **AsyncTask** umožňuje vykonávať operácie na oddelenom vlákne a tieto operácie bežia paralelne s hlavným vláknom. Po úspešnom získaní prístupového tokenu sa token uloží do jednej z premenných wrappera. Následne sa zavolá metóda **createAccount** a vytvorí nový objekt typu **Account**. Do tohto novo vytvoreného objektu sa vloží získaný prístupový token. Metóda **fillID** vloží do objektu typu **Account** unikátny identifikátor

---

užívateľa (serviceId). Nakoniec sa tu nachádza metóda `clearGPlusConnection`, ktorá odhlási užívateľa v prípade, že je prihlásený.

#### 6.4.5 Problémy pri implementácii

Metódy `getGroups` a `updateGroupInfo` v konektore nie sú momentálne funkčné. Google Plus API zatiaľ nepodporuje výpis zoznamu skupín, v ktorých máte uložených svojich priateľov. Očakávame, že v blízkej budúcnosti umožní Google vo svojom API túto funkcionality.

### 6.5 Twitter konektor a wrapper

#### 6.5.1 Princíp fungovania

Komunikácia môže prebiehať v dvoch režimoch. V prvom režime nie je potrebná autentifikácia užívateľa aplikácie. Týmto spôsobom sa dajú získať len verejne publikované informácie. Druhý režim používa protokol OAuth verzie 1. Pomocou tohto režimu sa posielajú autorizované požiadavky na službu Twitter. Postup získania špeciálneho prístupového tokenu a prístupového tajného tokenu je nasledovný. Užívateľ požiadava autorizčný server o request token. Autorizačný server presmeruje užívateľa na prihlasovací formulár a vyžiada si prihlasovacie údaje. Sú nimi login a príslušné heslo, ktorými sa užívateľ prihlasuje na službu Twitter. V prípade správne zadaných údajov server odošle aplikácii v odpovedi request token. Aplikácia následne požiadava zameniť tento request token za prístupový token a prístupový tajný token. V prípade, že je tento request token zhodný s tokenom, ktorý autorizčný server Twitteru prideliť užívateľovi aplikácie server odošle užívateľovi unikátny prístupový token a prístupový tajný token. Pomocou týchto dvoch tokenov môže klient posilať autorizované požiadavky na server Twitteru. [27]

Autorizačný server má nastavené limity pre rôzne druhy požiadaviek na jeden prístupový token, aby nedochádzalo k zahlteniu. Doba limitu je momentálne nastavená na 15 minút. Hodnoty týchto limitov môžete nájsť na stránke <https://dev.twitter.com/docs/rate-limiting/1.1/limits>.

Na rozdiel od služby Google Plus je posielanie autorizovaného požiadavku zložitejšie. V tejto službe hlavička `Authorization` obsahuje až 7 hodnôt. Tieto hodnoty sú `OAuth_consumer_key`, `OAuth_nonce`, `OAuth_signature`, `OAuth_signature_method`, `OAuth_timestamp`, `OAuth_token` a `OAuth_version` (informácie čerpané z [28]).

**OAuth\_consumer\_key** – je daná pri registrácii vašej aplikácie na stránke Twitteru.

**OAuth\_nonce** – je unikátna a generuje sa pri každom novom požiadavku. Twitter pomocou tejto hodnoty určuje či bola určitá požiadavka poslaná viackrát. Hodnota sa generuje pomocou base64 kódovania 32 bitov náhodných dát.

---

**OAuth\_signature** – zabezpečuje, že sa dáta nezmenia v priebehu posielania na autorizačný server. Hodnota je generovaná zo všetkých ostatných parametrov požiadavky a dvoch tajných hodnôt v priebehu prihlasovacieho algoritmu.

**OAuth\_signature\_method** – je vždy HMAC-SHA1, ktorú určuje Twitter

**OAuth\_timestamp** – určuje kedy bola daná požiadavka vygenerovaná. Má unixový formát. Twitter neakceptuje požiadavky, ktoré boli vygenerované dávno v minulosti.

**OAuth\_token** – hodnota, ktorú získa užívateľ aplikácie od autorizačného serveru Twitteru po úspešnej autentifikácii.

**OAuth\_version** – je vždy 1.0. Určuje verziu protokolu OAuth.

### 6.5.2 Twitter API

Pri implementácii na strane serveru sú použité knižnice twitter – async zo stránky <https://github.com/jmathai/twitter-async>. Je to implementácia oficiálnej Twitter REST API využívajúcej protokol OAuth. Tieto knižnice zabezpečujú autorizáciu užívateľa našej aplikácie na autorizačnom serveri Twitter. Ďalej dokážu vytvárať a posilať autorizované požiadavky na autorizačný server. Využívajú tri objekty, ktorými sú EpiCurl, EpiOAuth, EpiTwitter.

**EpiCurl** – má za úlohu posielanie požiadaviek na server Twitter pomocou CURL funkcií.

**EpiOAuth** – vytvára autorizované požiadavky a generuje všetky potrebné parametre v hlavičke Authorization.

**EpiTwitter** – integruje Twitter API a získava informácie z Twitteru.

Wrapper používa k autorizácii Java knižnicu Twitter4j. Táto knižnica má na starosti kompletnú autorizáciu pomocou protokolu OAuth a dokáže posilať autorizované požiadavky na server Twitteru. Odpovede prichádzajú vo formáte JSON.

### 6.5.3 Funkcie v konektore

Konštruktor konektoru požaduje dve hodnoty. Hodnotu consumer key a consumer secret, ktoré získame po registrácii našej aplikácie. Tieto hodnoty sa v priebehu komunikácie nemenia preto sú uložené po celú dobu v súkromných premenných v objekte konektoru. Pri volaní ostatných metód (validateAccount, getFriends, getGroups, atď.) sa ukladá do lokálnej premennej v danej metóde Twitter objekt podobne ako pri Google Plus konektore (zachovanie reentrantnosti konektoru). Tento objekt má na starosti komunikáciu medzi našim serverom a serverom Twitteru, z ktorého získava informácie. Objekt sa vytvára pomocou metódy constructTwitterObject, ktorá ako vstupný parameter požaduje objekt Account. V tejto metóde sa vytvára objekt typu EpiTwitter. Jeho konštruktor požaduje

---

hodnoty consumer key, consumer secret, OAuth token a OAuth token secret. Napokon sú tu ešte dve špeciálne vytvorené metódy konkrétne pre tento konektor. Sú to combineTokenAndSecret a getTokenAndSecret metódy. Ich účelom je spojiť hodnoty OAuth\_token a OAuth\_token\_secret do jednej a uložiť ich do objektu Account pod jedinou premennou access\_token. Postup je veľmi jednoduchý. Medzi tieto dve hodnoty, ktoré potrebujeme spojiť sa vloží znak @, keďže tento znak sa nepoužíva v týchto reťazcoch. Rozdelenie na dve hodnoty je taktiež jednoduché. Vyhľadá sa znak @ v reťazci. Časť reťazca pred týmto znakom sa uloží do jednej premennej a zvyšná časť do druhej premennej (znak @ sa ignoruje).

#### 6.5.4 Funkcie vo wrapperi

Konštruktor TwitterWrapperu je prázdny. Na začiatku sa deklarujú potrebné konštanty, v ktorých sú uložené údaje získané po registrácii našej aplikácie na stránke Twitteru. Metóda doLogin má na starosť autorizáciu a uloženie získaných údajov do spoločnej pamäti aplikácie. Je volaná na samostatnom vlákne a využíva objekt typu WebView, na ktorý je naviazaný objekt typu WebViewClient. Po detekcii špecifickej URL adresy tento WebViewClient objekt získa z adresy prístupový token a prístupový tajný token. Tieto dva tokeny sa následne uložia do spoločnej pamäti aplikácie. Metóda nazvaná createAccount sa volá po zápise prístupových údajov do spoločnej pamäti. Jej úlohou je prečítať prístupové tokeny z tejto pamäti a vložiť ich do novo vytvoreného objektu typu Account. Ďalšia v poradí je metóda clearTwitterConnection, ktorá sa volá pred autorizačným procesom a následne po ňom. Jej funkciou je vymazanie súčasných prístupových údajov z pamäti aplikácie a vytvorenie nového hlavného Twitter objektu s prázdny prístupovým tokenom. Nakoniec tu máme ešte metódu setToAccount, ktorá prečíta z objektu typu Account prístupové tokeny a nastaví podľa nich hlavný Twitter objekt. Metóda sa volá pri získavaní unikátneho identifikátoru pre objekt typu Account v metóde fillID.

### 6.6 LinkedIn konektor a wrapper

#### 6.6.1 Princíp fungovania

Po zaregistrovaní vašej aplikácie na oficiálnych stránkach LinkedIn máte možnosť využívať dve verzie protokolov OAuth. Verziu 1.0a a 2.0. Postačujúca a bežne používaná je práve verzia 1.0a. Autentifikačný a autorizačný proces prebieha podobne ako pri dvoch vyššie spomenutých sociálnych sieťach. Najprv si aplikácia vyžiada od autorizačného serveru LinkedIn autorizačný kód. Serveru pošle HTTPS request, v ktorom je uložený API key (získaný pri registrácii našej aplikácie) a náhodný reťazec v premennej state, ktorý slúži ako ochrana proti útokom typu Cross-site request forgery. Po úspešnom odoslaní tejto požiadavky server verifikuje zaslané údaje. Pokiaľ sú správne server pošle aplikácii dočasný autorizačný kód spolu s reťazcom state. Tento reťazec sa musí zhodovať s tým,

---

ktorý bol zaslaný v požiadavke na server. Po získaní autorizačného kódu aplikácia pošle ďalšiu požiadavku na server LinkedIn, v ktorej sú okrem tohto kódu opäť uložené hodnoty získané po registrácii aplikácie (hodnoty API key a API secret). Pokiaľ je požiadavka správne zostavená a hodnoty sú správne, potom autorizačný server pošle aplikácii prístupový token, ktorý je štandardne použiteľný po dobu 60 dní. Po získaní tohto tokenu už môžete pomocou vašej API posilať autorizované požiadavky na LinkedIn server, ktorý Vám bude vracat' požadované údaje. Musí byť však zabezpečené taktiež obnovenie prístupového tokenu ešte pred jeho expiráciou. Nakoniec treba ešte spomenúť takzvaný scope, podľa ktorého nastavujete aké časti vášho LinkedIn konta sa majú sprístupniť aplikácii. Sú to napríklad váš profil, vaša emailová adresa, vaši priatelia, skupiny a podobne. [29]

### 6.6.2 LinkedIn API

V tomto konektore je použitá voľne dostupná trieda, ktorá implementuje funkcie LinkedIn API. Je to vlastne jednoduchý a samostatný wrapper, v ktorom sú zakomponované tie najpoužívanejšie funkcie. Z tejto triedy sa vytvára objekt typu LinkedIn, ktorý má na starosť získavanie dát zo serveru LinkedIn a ďalej komunikuje s OAuth PHP knižnicou, ktorá je uložená v samostatnom súbore. [30]

Wrapper v Android aplikácii používa oficiálnu Android API od LinkedIn. Pomocou nej je možné vykonávať všetky potrebné operácie od autorizácie pomocou protokolu OAuth až po spracovanie odpovedí zo serveru LinkedIn.

### 6.6.3 Funkcie v konektore

Košírktor tohto konektoru požaduje hodnoty API key a API secret. Tie sa následne uložia do poľa apiConfig pomocou metódy setApiConfig, ktoré sa ďalej predá košírktoru objektu typu LinkedIn. Volanie klasických metód ako getPersonInfo, validateAccount alebo getFriends prebieha podobne ako pri GooglePlus alebo Twitter konektore. Najprv sa volá metóda constructLinkedInObject, ktorá vytvorí nový objekt typu LinkedIn. Do tohto objektu sa vloží vyššie spomínané pole apiConfig, nastaví sa prístupový token a JSON formát odpovede. Metóda vráti objekt typu LinkedIn, ktorý sa uloží do lokálnej premennej čím je zachovaná reentrantnosť konektoru. Ešte sú tu dve špeciálne metódy, ktoré pracujú s prístupovými tokenmi. Pretože server LinkedIn vracia aplikácii pole 4 hodnôt, ktoré spolu tvoria prístupový token. Tieto 4 hodnoty je potrebné vybrať z poľa a spojiť ich do jedného reťazca, aby bolo následne možné uloženie do objektu typu Account. Používa sa k tomu oddeľovací znak @ podobne ako pri Twitter konektore, pretože tento znak sa v hodnotách poľa nepoužíva. Týmto bola vysvetlená funkcia metódy combineAuthorizationInfo a nakoniec tu máme ešte metódu getAuthorizationInfo. Tá z objektu typu Account prečíta reťazec z premennej

---

accessToken a rozdelí ho do potrebných 4 hodnôt podľa oddeľovacích znakov @. Získané hodnoty následne vloží do autorizačného poľa.

#### 6.6.4 Funkcie vo wrapperi

LinkedIn wrapper je veľmi podobný Twitter wrapperu. Má prázdny konštruktor. Na začiatku definuje potrebné objekty pre prácu. Objekt typu LinkedInOAuthService má na starosť autorizáciu pomocou protokolu OAuth. LinkedInApiClient je hlavný LinkedIn objekt, cez ktorý sa posielajú požiadavky na server LinkedIn. Nájde tu podobne ako pri Twitteri metódy doLogin, createAccount a clearLinkedInConnection (v Twitter wrapperi metóda clearTwitterConnection). Metóda setToAccount sa tu nenachádza, ale jej funkcionality je implementovaná v metóde fillID. V metóde doLogin nájdeme objekt typu WebView, na ktorý je naviazaný objekt typu WebViewClient. Tu prebieha opäť detekcia určitej URL adresy, z ktorej sa následne prečítajú prístupové tokeny a uložia sa do spoločnej pamäti aplikácie. Po úspešnom uložení sa volá metóda createAccount, ktorá vytvorí nový objekt typu Account, prečíta prístupové tokeny z pamäti aplikácie. Tie sa následne uložia do novo vytvoreného Account objektu. Trochu pozmenená metóda fillID oproti TwitterWrapperu prečíta zo zadaného objektu typu Account prístupové tokeny, vytvorí nový hlavný LinkedIn objekt a pomocou požiadavky získa od LinkedIn serveru unikátny identifikátor užívateľa. Identifikátor následne uloží do objektu typu Account.

#### 6.6.5 Problémy pri implementácii

V konektore nefungujú metódy getGroups a updateGroupInfo, pretože LinkedIn API momentálne nepodporuje získanie skupín priateľov z užívateľského profilu. Potvrdili mi to aj na oficiálnej stránke LinkedIn API, kde som písal či existuje nejaký spôsob ako získať tieto informácie. Ich odpoveď bola, že momentálne to nie je možné, ale určite to oznámia vývojovému tímu. Tým pádom sa dá predpokladať, že v blízkej budúcnosti bude aj táto funkcionality implementovaná v LinkedIn API. [31]

---

## 7 Záver

Teoretická časť bakalárskej práce zoznamuje čitateľa s aplikáciou MeetMe a jej architektúrou. Ďalej predstavuje najpopulárnejšie sociálne siete súčasnosti. Zaoberá sa taktiež prepojením aplikácie s týmito sociálnymi sieťami a napokon popisuje implementáciu jednotlivých konektorov a wrapperov.

Praktická časť bola vytvorená po zoznámení sa s požiadavkami a architektúrou systému. Najprv som implementoval konektory na strane serveru. Tieto konektory boli otestované na webovej stránke, ktorú som vytvoril za účelom testovania funkcionality. Každý konektor mal za úlohu autorizovať prihlasujúceho sa užívateľa a následne uložiť údaje o jeho profile a profily jeho priateľov do databáze. Po schválení od pána vedúceho som začal pracovať na implementácii wrapperov v Android aplikácii. Na otestovanie mi pán vedúci vytvoril špeciálnu testovaciu aplikáciu, kde každý wrapper musel autorizovať užívateľa a vypísať jeho unikátny identifikátor a prístupový token. V oboch implementáciách som sa stretol s rôznymi problémami. Avšak vždy sa mi podarilo nájsť nejaké riešenie, či už išlo o vlastný nápad alebo mi pomohli ľudia v rôznych internetových diskusiách.

Na priloženom CD k bakalárskej práci nájdete zdrojové kódy konektorov, testovacej stránky pre konektory, wrapperov a testovaciu Android aplikáciu. V prípade záujmu vidieť zvyšné zdrojové kódy kontaktujte spoločnosť SCOVECO.

Bakalárska práca mi rozšírila teoretické a praktické skúsenosti týkajúce sa objektového programovania v jazyku PHP, platformou Android a v oblasti prepájania aplikácií s rôznymi API sociálnych sietí. Najviac znalostí som využil z predmetu Programovacie jazyky a z predmetu Vývoj informačných systémov, ktoré som absolvoval na VŠB-TUO.

V najbližších mesiacoch očakávam, že Google API a LinkedIn API rozšíri svoju funkcionality o získavanie informácií o skupinách, do ktorých sú triedení priatelia. Túto funkcionality by som chcel následne implementovať do konektorov. Potom by nasledovalo dôkladné otestovanie aplikácie MeetMe na rôznych verziách platformy Android, ktoré momentálne chýba. Po otestovaní už bude závisieť budúcnosť aplikácie MeetMe na firme SCOVECO.

---

## 8 Referencie

- [1] BERSON, A. *Client/Server Architecture*. 2nd ed.: McGraw - Hill, 1996. 569 p. ISBN 0070056641.
- [2] ZANDSTRA, M. *PHP 5 Objects, Patterns, and Practice*. 1st ed.: Apress, 2005. 438 p. ISBN 1590593804.
- [3] GERAGHTY, Bryan. Bryan Geraghty Blog [online]. 2012-07-16 [cit. 2013-05-04]. *Why I Choose PHP*. Dostupné z WWW:  
<<http://bryan.ravensight.org/2012/07/why-i-choose-php/>>
- [4] The Apache HTTP Server Project [online]. 2012 [cit. 2013-05-04]. *About the Apache HTTP Server Project*. Dostupné z WWW:  
<[http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html)>
- [5] SCHILDT, H. *Java, A Beginner's Guide*. 5th ed.: McGraw-Hill Osborne Media, 2011. 640 p. ISBN 0071606327.
- [6] MEIER, R. *Professional Android 4 Application Development*. 3rd ed.: Wrox, 2012. 864 p. ISBN 1118102274.
- [7] LAURENT, ST.S., DUMBILL, E., JOHNSTON, J. *Programming Web Services with XML-RPC*. 1st ed.: O'Reilly Media, 2001. 230 p. ISBN 0596001193.
- [8] Sociální síť - Wikipedie [online]. St. Petersburg (Florida) : Wikipedia Foundation, , [cit. 2013-05-04]. Dostupné z WWW:  
<[http://cs.wikipedia.org/wiki/Sociální\\_síť](http://cs.wikipedia.org/wiki/Sociální_síť)>
- [9] SCOTT, P.J. *Social Network Analysis*. 3rd ed.: SAGE Publications Ltd, 2012. 216 p. ISBN 1446209040.
- [10] Tweet Smarter: Official blog of @Twitter\_Tips [online]. 2012-05-15 [cit. 2013-05-04]. *Infographic: Spring 2012 Social Media User Statistics*. Dostupné z WWW:  
<<http://blog.tweetsmarter.com/social-media/spring-2012-social-media-user-statistics/>>
- [11] Facebook - Wikipedia, the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, , [cit. 2013-05-04]. Dostupné z WWW:  
<<http://en.wikipedia.org/wiki/Facebook>>
- [12] Google+ - Wikipedia, the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, , [cit. 2013-05-04]. Dostupné z WWW:  
<<http://en.wikipedia.org/wiki/Google+>>



- 
- [13] WOJDYLO, Jesse. Google+ [online]. 2012-12-06 [cit. 2013-05-04]. *Google+ Communities - Here Is Why It Will Change*. Dostupné z WWW:  
<<https://plus.google.com/116024884086268367178/posts/RMPeZyykKc9>>
- [14] Google+ : <https://plus.google.com/> [online]. [cit. 2013-05-04]. *Vytvořte událost – Google+*. Dostupné z WWW:  
<<http://www.google.com/+/learnmore/events/>>
- [15] Branded Clever - Web Design, Web Development, and Technology Solutions for Small Business. [online] 2011-07-01 [cit. 2013-05-04]. *Five Steps to Configuring Privacy on Google Plus (+)*. Dostupné z WWW:  
<<http://www.brandedclever.com/five-steps-to-configuring-privacy-on-google-plus/>>
- [16] Twitter - Wikipedia, the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, , [cit. 2013-05-04]. Dostupné z WWW:  
<<http://en.wikipedia.org/wiki/Twitter>>
- [17] Twitter Help Center : <http://support.twitter.com/> [online]. [cit. 2013-05-04]. *Using hashtags on Twitter*. Dostupné z WWW:  
<<http://support.twitter.com/entries/49309-what-are-hashtags-symbols>>
- [18] Twitter Help Center : <http://support.twitter.com/> [online]. [cit. 2013-05-04]. *What are @replies and mentions?* Dostupné z WWW:  
<<https://support.twitter.com/entries/14023-what-are-replies-and-mentions>>
- [19] QuickOnlineTips (QOT). [online] 2012-05-18 [cit. 2013-05-04]. *How to Protect Privacy on Twitter*. Dostupné z WWW:  
<<http://www.quickonlinetips.com/archives/2012/05/protect-privacy-on-twitter/>>
- [20] LinkedIn - Wikipedia, the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, , [cit. 2013-05-04]. Dostupné z WWW:  
<<http://en.wikipedia.org/wiki/LinkedIn>>
- [21] LinkedIn - Wikipedia, the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, , [cit. 2013-05-04]. Dostupné z WWW:  
<<http://en.wikipedia.org/wiki/LinkedIn#Groups>>
-

- 
- [22] CIO.com - Tech News, Analysis, Blogs, Video. [online] 2010-11-02 [cit. 2013-05-04]. *5 LinkedIn Privacy Settings You Need to Know*. Dostupné z WWW:  
<[http://www.cio.com/article/632322/5\\_LinkedIn\\_Privacy\\_Settings\\_You\\_Need\\_to\\_Know?page=1&taxonomyId=3119](http://www.cio.com/article/632322/5_LinkedIn_Privacy_Settings_You_Need_to_Know?page=1&taxonomyId=3119)>
- [23] JACOBSON, D., BRAIL, G., WOODS, D. *APIs: A Strategy Guide*. 1st ed.: O'Reilly Media, 2011. 150 p. ISBN 1449308929
- [24] IETF Tools [online]. 2012-10-01 [cit. 2013-05-04]. *The OAuth 2.0 Authorization Framework*. Dostupné z WWW:  
<<http://tools.ietf.org/html/rfc6749>>
- [25] OAuth - Wikipedia, the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, , [cit. 2013-05-04]. Dostupné z WWW:  
<[http://en.wikipedia.org/wiki/OAuth#OpenID\\_vs.\\_pseudo-authentication\\_using\\_OAuth](http://en.wikipedia.org/wiki/OAuth#OpenID_vs._pseudo-authentication_using_OAuth)>
- [26] Google Developers: <https://developers.google.com/> [online]. [cit. 2013-05-04]. *Using OAuth 2.0 to Access Google APIs*. Dostupné z WWW:  
<<https://developers.google.com/accounts/docs/OAuth2>>
- [27] Twitter Developers: <https://dev.twitter.com/> [online]. 2013-03-05 [cit. 2013-05-04]. *Implementing Sign in with Twitter*. Dostupné z WWW:  
<<https://dev.twitter.com/docs/auth/implementing-sign-twitter>>
- [28] Twitter Developers: <https://dev.twitter.com/> [online]. 2013-03-05 [cit. 2013-05-04]. *Authorizing a request*. Dostupné z WWW:  
<<https://dev.twitter.com/docs/auth/authorizing-request>>
- [29] LinkedIn Developer Network: <http://developer.linkedin.com/> [online]. [cit. 2013-05-04]. *Authentication*. Dostupné z WWW:  
<<http://developer.linkedin.com/documents/authentication>>
- [30] Google Code: <https://code.google.com/> [online]. [cit. 2013-05-04]. *Simple – linkedinphp – A PHP – based wrapper for the LinkedIn API*. Dostupné z WWW:  
<<https://code.google.com/p/simple-linkedinphp/>>
-

---

[31] LinkedIn Developer Network: <http://developer.linkedin.com/> [online]. [cit. 2013-05-04]. *Can you get your connection tag name with Connection API ? Dostupné z WWW:*

*<<http://developer.linkedin.com/forum/can-you-get-your-connection-tag-name-connection-api>>*

---

## 9 Prílohy

### 9.1 Popis registrácie aplikácie a stiahnutia Google API

Aby ste mohli využívať služby Google API musíte si najprv registrovať vašu aplikáciu. Táto registrácia prebieha na stránke <https://code.google.com/apis/console/>. V záložke Services si vyberiete, ktoré služby Google API chcete využívať. Okrem sociálnej siete Google Plus to môže byť napríklad služba Google Drive, Google Maps, atď.

Po výbere služby (služieb) sa prepnete do záložky API Access. V nej si musíte nechať vygenerovať nový API kľúč, ktorý vám zabezpečí prístup k zvoleným službám. Vygenerovanie je veľmi jednoduché. Vyplníte názov svojej aplikácie, webovú stránku projektu (voliteľné) a taktiež si môžete nahráť vlastné logo aplikácie. Po vygenerovaní API kľúča potrebujete vygenerovať kľúč pre klienta. Google API podporuje tri druhy klientov a každý má svoje špecifické nastavenia. V prípade, že vaším klientom bude webová aplikácia (webová stránka) stačí zadať iba URL adresu a callback adresu, kde bude presmerovaný užívateľ po úspešnej autorizácii a získaní prístupového tokenu. Takýto vygenerovaný klientský kľúč budete používať vo svojej aplikácii pri získaní prístupového tokenu. Tieto údaje s nikým nezdieľajte.

V prípade Android aplikácie je proces trochu zložitejší. Musíte vyplniť názov inštalačného balíčku vašej aplikácie a SHA1 hash kľúča, ktorým je podpísaná táto aplikácia. Používa sa k tomu program keytool nachádzajúci sa v priečinku kde máte nainštalovanú Javu. Príkazom `keytool -exportcert -alias androiddebugkey -keystore <cesta k debug alebo produkčnému keystroke> -list -v` získate SHA1 hash kľúča. Na testovacie účely môžete použiť súbor debug.keystore, ktorý nájdete v operačnom systéme Windows na nasledujúcom mieste `C:\Users\<meno užívateľa>\.android\debug.keystore`. V prípade nejakých nejasností prikladám odkaz na stránku, kde je vyššie spomenutý proces pekne popísaný <https://developers.google.com/console/help/>. Tento proces má jednu významnú výhodu. Svoj klientský kľúč si nemusíte pamätať, pretože sa v aplikácii nepoužíva. Namiesto neho sa porovnáva SHA1 hash aplikácie a SHA1 hash uložený v Google API Console.

Po úspešnom zaregistrovaní aplikácie si treba stiahnuť balíček Google API pre konkrétny programovací jazyk na stránke [https://developers.google.com/+downloads/](https://developers.google.com/+/downloads/). V každom takomto balíčku nájdete aj vzorové príklady, na ktorých je funkčnosť API demonštrovaná.

---

## 9.2 Popis registrácie aplikácie a stiahnutia Twitter API

Na rozdiel od Google API je proces registrácie aplikácie omnoho jednoduchší. Na stránke <https://dev.twitter.com/> sa prihlásite pomocou svojho Twitter účtu. V záložke My Applications kliknete na tlačidlo Create a new application. Tam následne vyplníte formulár, kde zadáte názov vašej aplikácie, popis, webovú stránku aplikácie a callback URL, ktorá má rovnakú úlohu ako pri Google API (presmerovanie po získaní prístupového tokenu). Po vytvorení aplikácie v záložke Details uvidíte dva dôležité údaje, ktoré sa používajú pri komunikácii. Sú to hodnoty Consumer key a Consumer secret. Tieto hodnoty s nikým nezdieľajte. Posielajú sa autorizačnému Twitter serveru, ktorý vám za ne vygeneruje prístupový token. Týmto je registrácia aplikácie kompletná.

Ďalším krokom je stiahnutie Twitter API balíčku v závislosti na programovacom jazyku. Každý takýto balíček obsahuje súbor knižníc, ktoré komunikujú s Twitter API a zabezpečujú získanie prístupového tokenu pomocou protokolu OAuth2. Zoznam týchto balíčkov a podporovaných programovacích jazykov nájdete na stránke <https://dev.twitter.com/docs/twitter-libraries>. Jediné čo musíte zmeniť v týchto balíčkoch je vyplnenie vášho consumer key a consumer secret, ktoré sú unikátne pre vašu aplikáciu. Podobne ako pri Google API aj tu každý balíček obsahuje vzorové príklady využívajúce jednotlivé knižnice.

## 9.3 Popis registrácie aplikácie a stiahnutia LinkedIn API

Registrácia aplikácie prebieha na stránke <https://developer.linkedin.com/>. Tento proces je veľmi podobný ako v prípade Twitter API. Prihlásite sa pomocou svojho LinkedIn účtu a zvolíte záložku API Keys. Tam kliknete na zelené plus a zobrazí sa vám formulár. LinkedIn požaduje trochu viac kontaktných informácií na rozdiel od Twitteru. Po vyplnení informácií sa vám vygenerujú API kľúče, ktoré nesmiete s nikým zdieľať a sú unikátne pre vašu aplikáciu. Podobne ako u iných sociálnych sietí aj tu sa tieto kľúče používajú pri generovaní prístupových tokenov.

Balíčky, ktoré implementujú komunikáciu s API musíte hľadať sami alebo si vytvoriť vlastný balíček. Ja osobne som použil balíček pre PHP zo stránky <https://code.google.com/p/simple-linkedinphp/> a balíček pre Android/Javu <https://code.google.com/p/linkedin-j/>. Obidva balíčky obsahujú vzorové príklady. Jediné čo v nich musíte zmeniť je doplniť vaše API kľúče k aplikácii, ktoré ste si nechali vygenerovať pri registrácii aplikácie. V prípade, žeby ste si chceli vytvoriť vlastný balíček na stránke <http://developer.linkedin.com/documents/authentication> nájdete kompletnú dokumentáciu, ktorá popisuje proces získania prístupového tokenu a následnú komunikáciu so serverom LinkedIn. Táto dokumentácia obsahuje aj vzorové kódy.

---

---